

THE DYNALINK SYSTEM

David Sommer, Optimal Systems, Inc.
David Goossen, DynaMark, Inc.
Andrew Traldi, Traldi, Inc.

INTRODUCTION

A common problem in many organizations is providing access to large data files for users with no programming experience. An ideal answer would:

- Require no special training.
- Allow flexible retrieval of data meeting complex criteria.
- Permit users to invoke either pre-defined or *ad hoc* reports.
- And do this quickly, using modest computing resources.

We present DynaLink as an example of a system that goes a long way towards these goals. DynaLink was built to satisfy requirements in the direct marketing industry, but it stands as an approach useful in other industries as well.

The direct mail industry has traditionally concentrated on reducing unit costs by efficiencies in letter printing, addressing, and packaging. With rising postal costs, and with the traditional methods mostly exhausted, competition must now concentrate on "smart mailing" - selection of recipients who are likely to produce profitable business for the client company.

Many client companies have extensive information about potential mail recipients, usually because of existing customer relationships. Using that information effectively has usually been a rather *ad hoc* process, involving the writing of many special programs to filter the mail files for optimal selection.

The SAS® system is used by many direct mail companies, mostly because the writing of *ad hoc* programs is quicker than with traditional languages, but also because of the extensive library of procedures. PROC FREQ and PROC TABULATE are very often used to examine the potential impact of a selection strategy.

A major frustration for direct mailing companies and their clients is the pressure to do the right selections in what is always too short a period of time. Ambiguous communications between client and direct mail firm; the difficulty of writing correct programs quickly; and the frequent discovery late in the processing cycle that the mail file does not have exactly the information expected all result in missed deadlines, rising costs, and non-optimal mail selection.

DynaMark, Inc., has developed a system, known as DynaLink, that permits the client company to directly and efficiently explore its mail files and perform mail selections. DynaLink dramatically reduces communication errors and eliminates much time-consuming *ad hoc* programming. Features of DynaLink that produce these advantages are:

- A system using SAS/AF® software that allows the user to specify selection choices, report requirements, and to initiate a batch job to perform the requested analysis.
- Within the user front-end, *decision tables* are the vehicle for expressing selection rules. A special-purpose editor to allow the user to create and maintain these tables is implemented in SAS/AF. This editor takes advantage of the extended table capability of SAS/AF, allowing for decision tables up to 600 lines in length.
- The batch portion of DynaLink parses one or more decision tables, creating SAS code to represent the choices and actions requested by the user. This SAS code is then "%INCLUDEd" to actually perform the requested work.
- To speed response, information files are available in "full" as well as sample form. Typically, 0.5% and 10% files are constructed. A client can therefore get the logic right with quick runs against the samples before a "full" production run. The reports from sample runs are scaled up to represent "virtual reality," so counts and other statistics

represent an approximation to what will be seen when a production run is done.

INTERACTIVE FRONT-END

SAS/AF was used to build an interactive front end. This portion of DynaLink allows a user to select a subsystem (customer data base), then to identify a project, then within that project to build or edit decision tables. Finally, the user may specify one or more decision tables to be used, plus reports to be generated, and this information is used to create a batch job. That job is then submitted to another machine (currently, an IBM mainframe running under DOS/VSE®). Following are screen snapshots, with notes, to illustrate a typical session.

On initial entry, a user sees a welcome screen which stays in place until the system is initialized. The initialization process involves setting the environment so that the user has access to any necessary files or datasets. The user then sees the subsystem selection screen (Figure 1). This screen is actually customized for each user, with only those subsystems for which he/she is authorized appearing.

```

SCREENAF: Subsystem Selection
13=Help 15=End 19=Up 20=Down 22=Quit System...

Please select subsystem by moving cursor to desired entry and
pressing the Enter key.

After you select a subsystem, you will then proceed to the
Select Project screen.

LIST OF AVAILABLE SUBSYSTEMS FOR DMPGM038

AKBAR INTERNATIONAL
DEMO1 - DYNALINK
DEMO2 - HIERARCHY
DEMO3 - RELATIONAL
EBBETTS DIRECT
FORSTYHE TELECOMM
JAMAICA TECH
LAE REPORTS
PYTHAGORAS INSTITUTE
XENOPHOEICS INC
  
```

Figure 1: Subsystem Selection

After selecting a subsystem, the user must select a project. Any existing projects are displayed, and the user may pick one or initiate a new project (Figure 2). To start a new project, we use the Newproj command.

```

Project Selection
13=Help 15=End 1  +NewProj.....+
Directory of pro  Enter project name: TEST
                  Enter description
Project Descript  ==> TEST PROJECT FOR SUGI DEMO
SUGI    SUGI YES
                  (OK)      (Cancel)
  
```

Figure 2: Project Selection

```

DEMO1 - DYNALINK SUGI    SUGI TEST PROJECT

Main Menu

Edit Tables  Reports

CMS Subsect: Select Project: File modify: Help

Utilities  System Menu

Place cursor on your selection and press the enter key.
  
```

Figure 3: Main Menu

After selection of subsystem and project, the Main Menu screen appears (Figure 3). This screen is the principal navigation tool for users. Usually, Edit Tables is selected when beginning a new project since there are no decision tables yet.

After selecting the Edit Tables option, we proceed to the next screen, where a decision table is selected (Figure 4). If there are existing tables, the user may choose one by positioning

Edit/Print Decision Tables 13=Help 15=End 19=Up 20=Dn			+Newtable...	
SYSTEM: DEMO1 - DYNALINK			Enter table name: TABLE1	
PROJECT: TEST TEST PROJ			Enter description	
TABLE CREATED MODIFIED			==> TABLE FOR DEMO SYSTEM	
			(OK) (Cancel)	

Figure 4: Table Selection

the cursor and pressing Enter; or, a new table can be started by using the Newtable command. In this case, since there are no existing tables, we must start a new one and enter a name and description for the table.

The Decision Table Screen allows up to 600 lines in a single table. When creating a new table, Dynalink gives the user 6 lines to work with (Figure 5). This is easily expanded using editing commands in the AC field similar to those available in the SAS Program Editor or XEDIT. We built our own editor, however, because within a table we have two levels: Tests and Lines. The TTT field gives the Test number, while LL gives the Line within the test. Details on Tables, Tests, etc., are given in a later section of this paper.

Decision Table Screen 13=H 15=End Cursor... 17=Top 18=Bot 19=Up 20=Dn 22=Q Newname			
System: DEMO1 - DYNALINK		Project: TEST	
Table: TABLE1		TABLE FOR DEMO1 SYSTEM	
Current size: 6		Maximum: 600	
AC TTT LL VARIABLE ---DESCRIPTION--- CN -----VALUE(S)---			
--- 001 01 ---			
--- 001 02 ---			
--- 001 03 ---			
--- 001 04 ---			
--- 001 05 ---			
--- 001 06 XXXXXXXX XXXXXXXXXXXXXXXXXXXX TT XXXXXXXXXXXXXXXXXXXX			

Figure 5: Decision Table Screen

Help screens are available throughout Dynalink. These can be accessed through a function key. Figure 6 shows a two-screen Help describing the use of the Table Editing screen.

Decision Table Screen (H)			
In this screen, you create a decision table by entering tests into the table. Each line in a test specifies a variable, description, condition, and value. The variable can either be a variable that exists in the database, or a user-defined variable. The description field is optional for user-defined variables; it is advisable to enter one so that you can easily identify your user variables. A variable which exists in the database already has a definition; you cannot modify it. You must then enter information in the condition (CN) field and the VALUE(S) field. If you enter an illegal value a message explaining the error will appear and the field will be highlighted. You should correct the field before you start to enter data in other lines of the table. You may use editing commands in the Action (AC) field to move, insert, copy, or delete lines from the table.			
If you enter a ? in a field, you will get a list of legal values. If you press the help key (PF13) while the cursor is in a field, you will get more detailed help for that field.			
A ? in the VARIABLE field takes you to the data dictionary. Up and Dn move up or down 1 screen. Top and Bot move to the top or bottom of the table. You may specify a certain number of lines to move up or down by selecting the Cursor command. The Newname command allows you to enter a new name for the table you are currently editing.			
When you are done with your table, issue the End command. If you are creating a new table, the table will be saved and you will return to the options menu. If you are modifying a table, you will see a pop-up window which asks you to Cancel, Rename, or Overwrite. Cancel takes you back to the Decision Table while Rename allows you to save the table under a different name. Overwrite will save the current table without changing the name (i.e. the original table will be overwritten). Use the Cancel command to leave the edit screen without saving the table. If you were creating a new table it will not be saved; if you were editing an existing table, any changes made will not be saved. You will always be asked to confirm the Cancel command if you have entered any new information in the table.			
(13=Help)	(15=End)	(19=Up)	(20=Dn)

Figure 6: Decision Table Help Screens

After exiting from Help, we return to the decision screen. We enter a "?" in the VARIABLE field. This will take us to the Data Dictionary screen where we can select a variable that is in the data base.

The Data Dictionary screen appears (Figure 7). Note the choices on the Menu Bar - Find and Rfind are used very extensively to locate variables either by name or description. (The Data Dictionary itself is a SAS dataset, arranged in alphabetical order.)

Data Dictionary Search 13=Help Find 15=End 16=Rfind 17=Top 18=Bot 19=Up 20=Dn 22=Q		
VARIABLE	TYPE	DESCRIPTION
CORE ANV	N	AUTO MKT VALUE/NEW VEH PU
CORE AOF	N	AMT OF FIRST ORDER
CORE AOH	N	AMT OF HIGH ORDER
CORE AUO	N	AUTO OWNER
CORE DBI	N	DATA BASE ID
CORE DOF	D	DATE OF FIRST ORDER
CORE DOH	D	DATE OF HIGH ORDER
CORE DST	C	DWELLING SIZE/TYPE
CORE EHI	N	ESTIMATED HH INCOME
CORE HOI	N	HOME OWNERSHIP INDICATOR
CORE LOB	N	LIFE ORDER BAL
CORE LOR	N	LENGTH OF RESIDENCE
CORE MOB	N	MAIL ORDER BUYER INDICATO
CORE MRI	N	MAIL RESPONDER INDICATOR
CORE MRS	N	MARITAL STATUS
CORE MTO	N	MYRCHCKR OWNER
CORE NIF	C	NIXIE FLAG
CORE NOA	N	NUMBER OF ADULTS IN HH

Figure 7: Data Dictionary

We have chosen the Find command from the menu bar. Note the pop-up to select search type (Figure 8 - actual screen

Data Dictionary Search 13=Help Find... 15=End 16=Rfind 17=To +Find...			Select search type	
VARIABLE	TYPE	DESCRIPTION		
CORE ANV	N	AUTO MKT VALUE/NEW VEH P	> Description Variable	
CORE AOF	N	AMT OF FIRST ORDER		
CORE AOH	N	AMT OF HIGH ORDER	Select search direction	
CORE AUO	N	AUTO OWNER	> Forward Only	
CORE DBI	N	DATA BASE ID	All Entries	
CORE DOF	D	DATE OF FIRST ORDER	Backward Only	
CORE DOH	D	DATE OF HIGH ORDER		
CORE DST	C	DWELLING SIZE/TYPE	Search for:	
CORE EHI	N	ESTIMATED HH INCOME	INCOME	
CORE HOI	N	HOME OWNERSHIP INDICATOR		
CORE LOB	N	LIFE ORDER BAL		
CORE LOR	N	LENGTH OF RESIDENCE		
CORE MOB	N	MAIL ORDER BUYER INDICAT	(OK) (Cancel)	
CORE MRI	N	MAIL RESPONDER INDICATOR		
CORE MRS	N	MARITAL STATUS		
CORE MTO	N	MYRCHCKR OWNER		
CORE NIF	C	NIXIE FLAG		
CORE NOA	N	NUMBER OF ADULTS IN HH		

Figure 8: Data Dictionary Screen

simplified to fit in available space). Here we have elected to search the Description (rather than the Variable) for the word INCOME. The first item containing "INCOME" in its Description is now at the top of the screen (Figure 9).

Data Dictionary Search 13=Help Find 15=End 16=Rfind 17=Top 18=Bot 19=Up 20=Dn 22=Q		
VARIABLE	TYPE	DESCRIPTION
CORE EHI	N	ESTIMATED HH INCOME
CORE HOI	N	HOME OWNERSHIP INDICATOR
CORE LOB	N	LIFE ORDER BAL
CORE LOR	N	LENGTH OF RESIDENCE
CORE MOB	N	MAIL ORDER BUYER INDICATO
CORE MRI	N	MAIL RESPONDER INDICATOR
CORE MRS	N	MARITAL STATUS
CORE MTO	N	MYRCHCKR OWNER
CORE NIF	C	NIXIE FLAG
CORE NOA	N	NUMBER OF ADULTS IN HH
CORE NOC	N	NUMBER OF CHILDREN IN HH
CORE NOP	N	NUMBER OF PERSONS IN HH
CORE OAI	N	OLDEST AGE IN HH
CORE OCC	N	OCCUPATION CODE
CORE POC	N	PRESNCE OF CHILDREN INDI
CORE RVO	N	REC VEH OWNER
CORE STA	C	STATE
CORE TOP	C	TYPE OF FIRST ORDER

Figure 9: Data Dictionary Screen

We select it by pressing Enter, then enter the CN and VALUE(S) fields with "GE" and "25000" (Figure 10). The meaning is obvious: select observations with the variable CORE_EHI GE 25000. As users get familiar with a data base, they will often enter data base variable names directly, rather

```

Decision Table Screen
13=H 15=End Cursor... 17=Top 18=Bot 19=Up 20=Dn 22=Q Newname

System: DEMO1 - DYNALINK      Project: SUGI
Table: TABLE1      TEST FOR SUGI PROJECT
Current size: 6      Maximum: 600
AC TTT LL VARIABLE ---DESCRIPTION--- CN -----VALUE(S)---

001 01 CORE_EHI ESTIMATED HH INCOM GE 25000
001 02
001 03
001 04
001 05
001 06 XXXXXXXX XXXXXXXXXXXXXXXXXXXX TT XXXXXXXXXXXXXXXXXXXX

```

Figure 10: Decision Table Screen

than going through the Data Dictionary screen. But whether entered directly or by search in the Data Dictionary, all data base variables are automatically checked against the data dictionary. Also, User Variables may be entered. These begin with C_ (Character) or N_ (Numeric). Figure 11 shows the first screen of a more complicated table.

```

Decision Table Screen
13=H 15=End Cursor... 17=Top 18=Bot 19=Up 20=Dn 22=Q Newname

System: DEMO1 - DYNALINK      Project: SUGI
Table: SELECTMB SELECT MAILBASE
Current size: 28      Maximum: 600
AC TTT LL VARIABLE ---DESCRIPTION--- CN -----VALUE(S)---

001 01 MH00_D00 DATE OF OFFER      CE 9001,9002,9003
001 02                             CC 9004,9005,9006
001 03                             CC 9007,9008,9009
001 04                             CC 9010,9011,9012
001 05 XXXXXXXX XXXXXXXXXXXXXXXXXXXX TC XXXXXXXXXXXXXXXXXXXX
002 01 MH00_D00 DATE OF OFFER      EQ 9009
002 02 MH00_D00 DATE OF OFFER      NE 9010
002 03 MH00_D00 DATE OF OFFER      NE 9011
002 04 C PREVPR PREV PROGRAM      RC FALL I
002 05 OUTPUT                      RC Y
002 06 XXXXXXXX XXXXXXXXXXXXXXXXXXXX TT XXXXXXXXXXXXXXXXXXXX
003 01 MH00_D00 DATE OF OFFER      NE 9009
003 02 MH00_D00 DATE OF OFFER      EQ 9010
003 03 MH00_D00 DATE OF OFFER      NE 9011
003 04 C PREVPR PREV PROGRAM      RC FALL II

```

Figure 11: Decision Table Screen

When a user is done editing a table, it is saved in a SAS dataset on disk and we return to the main options menu. We will now demonstrate how a Mail Select is done. Using a previously defined project, we proceed to the main options menu (Figure 12). Note that two additional options are available: Print Tables and Mail Select. We choose the Mail Select option, which allows the user to designate one or more of his decision tables as logic to determine a mailing list.

DEMO1 - DYNALINK SUGI SUGI TEST PROJECT

Main Menu

Edit Tables

Print Tables

Mail Select

Reports

CMS Subset

Select Project

File modify

Help

Utilities

System Menu

Place cursor on your selection and press the enter key.

Figure 12: Main Menu

In the next screen, the user enters the parameters for the job (Figure 13). After entering the name and description, the user may choose between Virtual and real processing and then select an input source, which can either be the database or a previously created mail select file. This flexibility in input file selection gives the user the ability to fine tune the logic in the tables before running off the full database.

Next, the user selects which tables to include and in which order (Figure 14). In addition, the user can set additional parameters for the job by using the Profile command. These parameters include the detail level of reports, the number of

observations to process, the destination of the report, etc. Once all the tables are selected, the job is submitted. Dynalink builds

```

Job parameter selection
13=Help 15=End 19=Up 20=Dn 22=Quit System...
Select a database file
SUBSYSTEM: DEMO1 - DYNALINK      REPORT

Project: SUGI
Description: SUGI TEST PROJECT

Job Name: JDM TEST (? for list of existing jobs)
Description: MAIL SELECT: SUGI TEST PROJECT

(V)irtual=<default> or (R)real factor: V

AVAILABLE DATABASE FILES
5.0 PERCENT SAMPLE

```

Figure 13: Job Parameters Screen

the job stream, including the necessary JCL, and sends it to VSE, where the batch processing occurs.

```

Job Submission Screen
13=Help 14=Profile 15=End Jobparms... 19=Up 20=Dn 22=Quit

System: DEMO1 - DYNALINK      Project: SUGI      Job: JDMTEST1
Job Title: PROC MEANS REPORT
Proj. Desc: SUGI TEST PROJECT
File: 5.0 PERCENT SAMPLE      Input Mail Select:
DISK Output File:              VFACTOR: 020
# of records: MAX              Detail Level: 2      ROUTE: Q
Max obs. per record: 1000
SM TABLE  CREATED  MODIFIED  DESCRIPTION
SELECTMB 01/28/92 01/30/92 SELECT MAILBASE
MKTCEL   01/28/92 01/30/92 NTH MARKET CELLS

```

Figure 14: Job Submission

One powerful feature in Dynalink is the ability for the user to design formats for reports. The format design screen (Figure 15) is reached by choosing the Utilities option from the Main Menu, and then the Design Formats option from the Utilities screen.

```

User-designed formats
13=H 14=Sort 15=End 17=Top 18=Bot 19=U 20=D 22=Q 23=Bot 24=Sw
Enter information or Retrieve previously saved format.
SYSTEM: DEMO1 - DYNALINK
PROJECT: SUGI SUGI TEST PROJECT

Name of format (character formats must begin with $): _____
Type of format (E - Equality, R - Range): _____
Description: _____

AC          FORMATTED VALUE      DESCRIPTION
-----

```

Figure 15: User-designed Formats

In this screen, the user can define a format that can be used in *ad hoc* reports. For example, suppose that a format for the level of income was desired. This format could contain ranges which were meaningful to a particular marketing study. The user may enter formatted values if desired, but Dynalink automatically generates a descriptive formatted value which is used if the user does not enter one. See Figure 16 for an example of a user-designed format. Dynalink allows the user to create character or numeric formats and to choose between range and equality formats.

Once this format is saved, it can be accessed through any *ad hoc* or customized report that the user creates. A directory of available formats for a system is made available to the user; they may be deleted, renamed, viewed, or selected from this directory. This enables the user to see which formats are available while creating a report.

Other features of Dynalink include the capability to maintain a library of mail select files, which can be used as input for reports or other mail select jobs. Custom SAS code can be used to generate reports and macros can be invoked through the decision tables. The user-friendly environment of an interactive system and the straight forward design of the decision tables allows users with little or no programming experience to extract meaningful information for marketing research.

User-designed formats		
13=B 14=Sort 15=End 17=Top 18=Bot 19=U 20=D 22=Q 23=Ret 24=Sv		
Enter upper limits and formatted values (optional).		
SYSTEM: DEMO1 - DYNALINK		
PROJECT: SUGI SUGI TEST PROJECT		
Name of format (character formats must begin with \$): INC10K		
Type of format (E - Equality, R - Range) : R		
Description: HOUSEHOLD INCOME IN 10K (THEN BIGGER INCREMENTS)		
NUMERIC		
AC UPPER LIMIT	FORMATTED VALUE	DESCRIPTION
0	ILLEGAL VALUE	LOW<=0
10000		0<=10000
20000		10000<=20000
30000		20000<=30000
40000		30000<=40000
50000		40000<=50000
75000		50000<=75000
100000		75000<=100000
200000		100000<=200000
HIGH	OVER 200,000	200000<=HIGH

Figure 16: User-designed Formats

SYSTEM DETAILS

The key to the DynaLink system is the Table, in which a user with little or no programming experience is able to create complicated selection logic. [Historical note: the DynaLink tables are a variant of the Decision Tables in vogue in the late 1970's as a program specification and documentation device. We permit mixed Decision and Action lines within a Test. We simplified the table design so that only one Test appears in the width of the screen.]

As you see it on the screen a typical Line might look like this (heading line included for reference):

```
AC TTT LL VARIABLE ...DESCRIPTION.... CN ...VALUES.....
002 01 MH00_D00 DATE OF OFFER EQ 9009
```

The meaning of each field:

AC The ACTION field. Used to enter editing commands, such as "15" - Insert 5 lines.

TTT Test number. This is automatically generated by the system. Tells what Test this line is part of in the current Table.

LL Line number, also automatically generated. Gives the line number within the current Test.

VARIABLE A variable name. This can either be a variable found in the Data Dictionary (which is the case with MH00_D00), or it can be a User Variable (described later).

DESCRIPTION A description of the variable. This is automatically retrieved from the Data Dictionary in the case of database variables, or it may be entered by the user in the case of user variables.

CN A Condition. Often this specifies a logical comparison (here, "EQ" means to test whether MH00_D00 is equal to "9009"). But the CN field can also specify operations such as Nthing, Replacement, or Count. And also the CN field is used to mark the end of a Test (with a TT or TC).

VALUE(S) Usually a constant, such as "9009", but for some commands several constants may appear, separated by commas. Also in a few commands a variable may appear.

Once a user has built one or more decision tables, combined them into a job, and told the system to execute that job, what will happen? The job is sent to the batch part of the system, which retrieves information from the sample specified. Here's the way the job is processed:

1. Retrieve all necessary information for the next observation.
2. Starting with the first line of the first test of the first table, perform each operation (logical comparison, Nthing, etc., in order).

a. For logical comparisons, if the condition is "true", then go on to the next line. But if it is "false", then the current Test stops for this observation, and execution continues with the next Test in the current Table (or with the next Table if this is the last Test in the current Table).

b. Some operations such as Counts (CE and CR) perform some action but do not cause the current Test to end.

c. When the end of a Test is reached (indicated by a TT or TC in the Condition field), then if the Condition field is TT, the Table Terminates, meaning execution goes on to the next Table. But if a TC, then the Table Continues, meaning execution continues with the next Test in the current Table.

3. When all Tables have been executed for the current observation, then the information may be saved to an output dataset (if this is a Mail Select type of run). Whether it is actually saved or not depends on the value of a special variable named "OUTPUT". the user can control whether the observation is output to a result dataset by changing the value of OUTPUT to Y.
4. Continue with the next observation, that is, return to step 1. above. Do this until all observations in the chosen sample have been "passed" through the decision logic.

Before going on with details, let's look at part of a decision table and understand what it does (Figure 17).

Line 001 01 is simply a comment. 001 02 is a command that will display how many times the variable C_PREVPR is equal to each of the values "FALL I", "FALL II", "FALL III" or blank. 001 03 marks the end of a test - in this case, a "TC" meaning Table Continue. Lines 002 02 through 002 04 are Not Equal tests against each of these values, so only observations where

C_PREVPR is Not Equal to any of those values will reach the end of the test. Any such observation will then hit the TT (Table Terminate) command. Thus, the observations going on to test 003 are those for which C_PREVPR was equal to one of "FALL I" (etc.). 004 02 confirms that C_MKTCEL (Market Cell) is equal to blank. 004 03 again calls for the display of the values of C_PREVPR. Then at 004 04 a Date of Order variable

is tested to be sure it is not missing (meaning that an order was placed). If so, C_MKTCEL (the Market Cell variable) is set equal to 001. This table continues, setting the C_MKTCEL variable equal to 002 in 50000 instances if its an observation where mailing took place is "FALL I", but the addressee did not respond, etc. (On the following page is an "Nthing Report" which shows more of this table.)

To understand the Nthing Report, the XE code needs explanation. XE permits a pseudo-random split - some observations getting one value for the specified variable, some a second, etc. XE is useful in direct marketing because one can set up "market cells" with it. These are randomly drawn samples, each of which gets a different treatment - different offer, different packaging, different timing, etc. In the example shown at line 005 05 (see Figure 18), 50000 observations will have C_MKTCEL set equal to 002, and the remainder equal to missing.

```
001 01 ** COUNT PREVIOUS PROGRAM FLAG
001 02 C PREVPR PREV PROGRAM CE FALL I,FALL II,FALL III,$$BLANK
001 03 ***** TC *****
002 01 ** DROP NOT MAILED FALL I,II,III
002 02 C PREVPR PREV PROGRAM NE FALL I
002 03 C PREVPR PREV PROGRAM NE FALL II
002 04 C PREVPR PREV PROGRAM NE FALL III
002 05 ***** TT *****
003 01 C_PREVPR PREV PROGRAM CE FALL I,FALL II,FALL III,$$BLANK
003 02 ***** TC *****
004 01 ** CELL 001 PREVIOUS RESPONDERS
004 02 C MKTCEL MARKET CELL EQ $$BLANK
004 03 C_PREVPR PREV PROGRAM CE FALL I,FALL II,FALL III,$$BLANK
004 04 R500 D00 DATE OF ORDER NE $SMISSING
004 05 C MKTCEL MARKET CELL RC 001
```

Figure 17: Part of Table

NTHING REPORT. PROJECT: SUGI PROJ TITLE: SUGI TEST MAIL SELECT

SAMPLE: 5.0 PERCENT SAMPLE FREQ WEIGHT: 020 QRS FOUND: 240,180 OUT OF MAX

MAIL SELECT IN: MAIL SELECT OUT: UDMSUGI.DMPGM003.DMI

TABLE NAME -MKTCELL TABLE DESCRIPTION -SELECT AND CREATE MARKET CELLS

TTT	LL	VARIABLE	DESCRIPTION	CN	VALUE(S)	COUNT PASSING	STANDARD ERROR	VALUE	COUNT OF VALUE	PERCENT
000	00	MKTCELL	TABLE		SELECT AND CREATE MARKET CELLS	1,165,980	45			
001	00		--- START TEST ---			1,165,980	45			
001	01			**	COUNT PREVIOUS PROGRAM FLAG	1,165,980	45			
001	02	C_PREVPR	PREV PROGRAM	CE	FALL I, FALL II, FALL III, \$SBLANK	1,165,980	45	FALL I	292,560	25.09
001	02							FALL II	236,020	20.24
001	02							FALL III	187,620	16.09
001	02							\$SMISSING	449,780	38.58
001	03	XXXXXXXXXXXXXXXXXXXX	TC	XXXXXXXXXXXXXXXXXXXXXXXXXXXX		1,165,980	45			
002	00		--- START TEST ---			1,165,980	45			
002	01			**	DROP NOT MAILED FALL I, II, III	1,165,980	45			
002	02	C_PREVPR	PREV PROGRAM	NE	FALL I	873,420	2,094			
002	03	C_PREVPR	PREV PROGRAM	NE	FALL II	637,400	2,404			
002	04	C_PREVPR	PREV PROGRAM	NE	FALL III	449,780	2,351			
002	05	XXXXXXXXXXXXXXXXXXXX	TT	XXXXXXXXXXXXXXXXXXXXXXXXXXXX		0	45			
003	00		--- START TEST ---			716,200	2,351			
003	01	C_PREVPR	PREV PROGRAM	CE	FALL I, FALL II, FALL III, \$SBLANK	716,200	2,351	FALL I	292,560	40.85
003	01							FALL II	236,020	32.95
003	01							FALL III	187,620	26.20
003	02	XXXXXXXXXXXXXXXXXXXX	TC	XXXXXXXXXXXXXXXXXXXXXXXXXXXX		716,200	2,351			
004	00		--- START TEST ---			716,200	2,351			
004	01			**	CELL 001 PREVIOUS RESPONDERS	716,200	2,351			
004	02	C_MKTCEL	MARKET CELL	EQ	\$SBLANK	716,200	2,351			
004	03	C_PREVPR	PREV PROGRAM	CE	FALL I, FALL II, FALL III, \$SBLANK	716,200	2,351	FALL I	292,560	40.85
004	03							FALL II	236,020	32.95
004	03							FALL III	187,620	26.20
004	04	BS00	DOO DATE OF ORDER	NE	\$SMISSING	40,180	881			
004	05	C_MKTCEL	MARKET CELL	RC	001	40,180	881			
004	06	C_PREVPR	PREV PROGRAM	CE	FALL I, FALL II, FALL III, \$SBLANK	40,180	881	FALL I	17,620	43.85
004	06							FALL II	13,080	32.55
004	06							FALL III	9,480	23.59
004	07	XXXXXXXXXXXXXXXXXXXX	TC	XXXXXXXXXXXXXXXXXXXXXXXXXXXX		40,180	881			
005	00		--- START TEST ---			716,200	2,351			
005	01			**	CELL 002 FALL I MAILING ONLY	716,200	2,351			
005	02	C_MKTCEL	MARKET CELL	EQ	\$SBLANK	676,020	2,384			
005	03	C_PREVPR	PREV PROGRAM	CE	FALL I, FALL II, FALL III, \$SBLANK	676,020	2,384	FALL I	274,940	40.67
005	03							FALL II	222,940	32.98
005	03							FALL III	178,140	26.35
005	04	C_PREVPR	PREV PROGRAM	EQ	FALL I	274,940	2,050			
005	05	C_MKTCEL	MARKET CELL	XE	50000=002,T=270000	274,940	2,050	002	50,000	18.19
005	05							\$SMISSING	224,940	81.81
005	06	C_MKTCEL	MARKET CELL	EQ	002	50,000	978			
005	07	C_PREVPR	PREV PROGRAM	CE	FALL I, FALL II, FALL III, \$SBLANK	50,000	978	FALL I	50,000	100.00
005	08	XXXXXXXXXXXXXXXXXXXX	TC	XXXXXXXXXXXXXXXXXXXXXXXXXXXX		50,000	978			
006	00		--- START TEST ---			716,200	2,351			

CONFIDENCE INTERVALS ARE "COUNT PASSING" +/- Z*STANDARD ERROR, WHERE Z AND CORRESPONDING PROBABILITY P ARE AS FOLLOWS:
 Z=1.0, P=.682; Z=2.0, P=.955; Z=3.0, P=.997; Z=1.645, P=.90; Z=1.96, P=.95; Z=2.58, P=.99; Z=3.29, P=.999

Figure 18: Nthing Report

A number of reports are available in Dynalink, but the Nthing Report is the most frequently used. It allows a user to see in a very direct way the consequences of his/her selection logic. The report is designed so that results on a sample file (here, a five percent sample) are projected as if the run had been against the full file. Thus, the Count Passing and the Count of Value columns are scaled up (here, by a factor of 20) to represent the approximate counts one would have seen in a full run. Consider a few examples:

At line 002 02, a Not Equal test is displayed. The Count Passing is shown as 873,420. (Really, in the five percent run, the actual count was $873,420 \times .05 = 43,671$.) We speak of 873,420 as the *virtual* count passing. Its expected accuracy is shown by the Standard Error value of 2,094. A footnote explains how to interpret these values. For example, the confidence interval $873,420 \pm 2 \times 2,094$, or $873,240 \pm 4,188$, covers the number passing this test in the full file with probability .955. (This comes from the "Z=2.0, P=.955" in the footnote.) A second example: at 001 02 is a Count Equals operation. This tallies the number of times that C_PREVPR is equal to each of the values "FALL I", "FALL II", "FALL III", and blank. Since no decision is made by this operation, the Count Passing of 1,165,980 is the same as on the previous line. But the Count of Value shows the virtual counts for each of the four possibilities. A percent is also shown.

The Condition codes available in Dynalink fall in five categories: Comparison, Counting, Nthing, Replacement, and Test End. There are also comments (**) and continuation (CC).

COMPARISON

LE, LT, GE, GT: These accomplish Less than or Equal, Less Than, Greater than or Equal, and Greater Than comparisons respectively. The Variable is compared against a single con-

stant. Example:

```
VARIABLE ...DESCRIPTION... CN ...VALUES...
N_MYVAR NUMERIC VARIABLE LE 6
(Passes this condition if N_MYVAR is 6 or less.)
```

EQ: This comparison is "true" if the variable is equal to any of the constants given. Example:

```
VARIABLE ...DESCRIPTION... CN ...VALUES...
C_PREVPR PREV PROGRAM EQ FALL I, FALL II, FALL III
```

NE: This comparison is the opposite of EQ - it is "true" if the variable is Not Equal to any of the constants given.

RG: This comparison is "true" if the value of the variable falls in the range between the first and second constants given. Example:

```
VARIABLE ...DESCRIPTION... CN ...VALUES...
CORE_HIN HOUSEHOLD INCOME RG 35000,65000
```

This RG is true if CORE_HIN is in the inclusive range from 35000 to 65000.

COUNTING

The CE operation has already been discussed. There is also a CR (Count Range) operation. Example:

```
VARIABLE ...DESCRIPTION... CN ...VALUES...
CORE_HIN HOUSEHOLD INCOME CR 0,10000,25000,60000
```

The Nthing Report will show how many households have income ≤ 0 (including missing); > 0 but ≤ 10000 ; > 10000 but ≤ 25000 ; > 25000 but ≤ 60000 ; and > 60000 .

NTHING

Assignment to market cells is done via the Nthing operations XE and XP. In each case a user variable receives a value indicating which market cell this observation is assigned to. XE has been illustrated already; here's an example of XP:

VARIABLEDESCRIPTION... CNVALUES.....
 N_MYCELL MY MXT CELL XP 15=1,25=2,20=3
 This assigns the value 1 to 15% of the observations, the value 2 to 25%, the value 3 to 20%, and 'missing' to the remainder (40%).

Assignment of values is done in such a way that each value will be spread uniformly through all observations. Thus, the assignment is not random, but is actually deterministic. From a statistical point of view this may seem unusual, but the nature of direct marketing files makes this adequately random and the results follow the desired proportions very closely.

REPLACEMENT

Three replacement operations are available: RC, RV, and RX. RC is used frequently; RV and RX seldom. All are used to set or replace a value in a user variable. Examples:

VARIABLEDESCRIPTION... CNVALUES.....
 C_GOOD BEST ANSWER RC STATUE OF LIBERTY
 N_GROUP GROUP NUMBER RV DBGROUP
 N_RANDOM RAND BETW 1 AND 10 RX CEIL(10*РАНУНИ(0))

The C_GOOD variable receives the value "STATUE OF LIBERTY" in the first statement. In the second, N_GROUP gets the value contained in the variable DBGROUP (RV means "replace with variable").

Use of RX ("Replace with eXpression") should be used only by those familiar with datastep statements in SAS. The VALUES field contains any legal SAS expression. Note that the usual SAS closing semicolon is omitted. In this example, N_RANDOM will receive a random integer between 1 and 10.

TEST END

There are two ways to end a test. The usual is with a TT in the CN field. This has the effect that if all conditions in the test are met, the test is said to "pass" for the observation, and all remaining tests in the current table are skipped.

In some instances a TC test end is needed. When this is used, then even if an observation "passes" all conditions in the test, the next test is executed.

CONCLUSION

DynaLink is an example of a solution to a common problem: how to allow non-programmers to access large data files, where the inquiries may be complex, and where quick response is important. A design consideration for DynaLink has been that inevitably, complex retrievals require "debugging." Hence, we have used sample files as a test-bed, and made all reports represent a "virtual reality" by appropriate scaling and the use of simple standard error information.

DynaLink has been in client use since July, 1991. Client response has been very positive. The goals of improving quality and timeliness of mail selection are consistently met.

NOTICE

SAS and SAS/AF are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration. DOS/VSE is a registered trademark or trademark of International Business Machines Corporation.

AUTHORS

David Sommer is president of Optimal Systems, Inc., 10338 Colorado Rd, Bloomington, MN 55438. 612-831-5724.

David Goossen is employed by DynaMark, Inc., 4670 Churchill St, St. Paul, MN 55126. 612-482-8593.

Andrew Traldi is president of Traldi, Inc., 2701 Spinnaker Court, Duluth, GA 30136. 404-418-0327.